



Universidade Federal de Alagoas (UFAL)
Campus Arapiraca



Programação Orientada a Objetos (POO)

01 - Descrição da disciplina

Alexandre de Andrade Barbosa

alexandre.barbosa@arapiraca.ufal.br

Objetivos

Objetivos

- Apresentar como será conduzida a disciplina
- Descrever brevemente os tópicos que serão trabalhados na disciplina

Sobre a disciplina

- Horário
- A disciplina na WEB
- Ementa
- Objetivos
- Conteúdo programático
- Metodologia de ensino
- Metodologia de avaliação
- Dúvidas frequentes
- Bibliografia

Horário

	Seg.	Ter.	Qua.	Qui.	Sex.
07:30-08:20	ES1	Plan./Atend.	Plan./Atend.	Pesq./Ext.	Pesq./Ext.
08:20-09:10	ES1	P1	LP1	Pesq./Ext.	Pesq./Ext.
09:10-10:00	Plan./Atend.	P1	LP1	Pesq./Ext.	Pesq./Ext.
10:20-11:10	Plan./Atend.	P1	LP1	Pesq./Ext.	Pesq./Ext.
11:10-12:00	Plan./Atend.	Reunião	Plan./Atend.	Pesq./Ext.	Pesq./Ext.
12:00-13:30	- -	- -	- -	- -	- -
13:30-14:20	POO	Plan./Atend.	Plan./Atend.	Pesq./Ext.	Pesq./Ext.
14:20-15:10	POO	Plan./Atend.	Plan./Atend.	Pesq./Ext.	Pesq./Ext.
15:20-16:10	Plan./Atend.	POO	SD	Pesq./Ext.	Pesq./Ext.
16:10-17:00	Plan./Atend.	POO	SD	Pesq./Ext.	Pesq./Ext.
17:00-18:00	Plan./Atend.	Plan./Atend.	SD	Pesq./Ext.	Pesq./Ext.

A disciplina na WEB

- **Página da disciplina:** <https://ava.ead.ufal.br>
- Avisos, conteúdo das aulas, bibliografia, dúvidas, submissão de trabalhos

Ementa

Ementa de POO no PPC:

- Conceitos de programação orientada a objetos
- Classes, objetos, atributos, operações, construtores e mensagens
- Tipos de dados
- Abstração
- Encapsulamento
- Associação e composição de objetos
- Herança, polimorfismo e ligação dinâmica
- Coleções de objetos
- Tratamento de erros usando exceções
- Introdução aos padrões de projetos

Objetivos

Objetivos de acordo com o PPC:

- Descrever os conceitos do paradigma de programação de orientação a objetos
- Capacitar os discentes na utilização da orientação a objetos utilizando uma linguagem de programação
- Ao final da disciplina o aluno será capaz de:
 - compreender e modificar programas orientados a objetos;
 - implementar programas em uma linguagem de programação orientada a objetos.

Conteúdo programático

1ª Parte (AB1)

- Programação estruturada em Java
- Tipos de dados
- Abstração
- Classes, objetos, atributos, operações, construtores e mensagens
- Encapsulamento (modificadores)
- Associação e composição de objetos
- Coleções de objetos

Conteúdo programático

2ª Parte (AB2)

- Herança, polimorfismo e ligação dinâmica
- Classes abstratas e Interface
- Tratamento de erros usando exceções
- Introdução aos padrões de projetos

Conteúdo programático

Tópicos adicionais (*/**)

- Interface gráfica com o usuário
- Conexão com banco de dados
- Aplicações móveis
- Aplicações web
- Desenvolvimento de jogos

* Não serão avaliados

** Serão apresentados apenas ao final do conteúdo regular

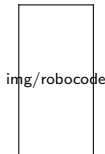
Metodologia de ensino

- Aulas expositivas (Sala de aula)
- Aulas práticas (LECC 1)
- Listas de exercícios
- Mini projeto

Metodologia de avaliação

- Listas de exercícios (podem valer pontos de salvação)
- Mini projeto (podem valer pontos de salvação)
- Provas escrita (foco em aspectos teóricos e práticos)
- Projetos (foco em aspectos práticos)

Metodologia de ensino e avaliação



- Mini projeto: Torneio Robocode (INDIVIDUAL)
- Forma de disputa: Fase de classificação e eliminatórias
- Cada vitória valerá alguma pontuação
- Robôs que obtenham ao menos uma vitória deverão ser apresentados

Metodologia de avaliação

- AB1
 - Prova 1 = 35%
 - Prova 2 = 35%
 - Projeto (Etapa 1) = 30%
- AB2
 - Prova 3 = 35%
 - Prova 4 = 35%
 - Projeto (Etapa 2) = 30%
- Reposição - Prova escrita + extensão do deadline do projeto
- Final - Prova escrita

Metodologia de avaliação

Provas

Provas

- Data marcada
- Todo conteúdo anterior ao dia da prova
- Escrita
- Sem consulta
- Individual

Metodologia de ensino e avaliação

Provas

Provas

- Provas compostas de por no máximo 10 questões (objetivas e/ou abertas com/sem implementação)
- Muitas questões do ENADE e Poscomp serão utilizadas
- As questões de prova **NÃO** estão restritas ao conteúdo dos slides, leiam as “leituras recomendadas” e materiais adicionais indicados no ava
- Provas serão agendadas e terão tolerância de 15 min. para atrasos*
- Não será permitido uso do celular, ou qualquer outro eletrônico
- Nenhum tipo de material deverá ser compartilhado sem autorização do professor
- Sob nenhuma hipótese será aplicada segunda chamada para as avaliações **

Metodologia de avaliação

Projetos

Projeto

- 1ª Entrega:
 - Protótipo executável
- 2ª Entrega:
 - Programa completo e artefatos associados

Metodologia de ensino e avaliação

Projeto

- Equipes: no máximo 3 alunos
- Entrega: através do AVA
- Data de entrega: agendada no AVA
- Público: apresentação apenas para o professor
- Apresentador: TODOS os integrantes da equipe
- Avaliações serão INDIVIDUAIS e baseadas em:
 - Utilização dos conceitos trabalhados na disciplina (**orientação a objetos**)
 - Respostas adequadas para as perguntas

Metodologia de avaliação

Ferramentas

Oracle JDK versão 11

```
java version "11.0.2" 2019-01-15 LTS
```

```
Java(TM) SE Runtime Environment 18.9 (build 11.0.2+9-LTS)
```

```
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.2+9-LTS, m
```

Sistema: TheHuxley (Obs.: openJDK 1.7)

Deepin 15.10 (sistema operacional)

Eclipse for Java Developers (IDE preferencial)

Dúvidas Frequentes

- Posso faltar quantas vezes?
 - As normas da UFAL descrevem que são tolerados 25% de faltas. As faltas serão inseridas no sistema semanalmente.
- Faltei a prova, posso realizar uma segunda chamada?
 - Você deve realizar a reposição correspondente.
 - Casos específicos tem direito a segunda chamada (consulte as normas da UFAL)
- Como entrego os trabalhos?
 - Nunca de forma impressa, sempre por meio digital, através do sistema (<https://ava.ead.ufal.br>) ou por email

Bibliografia básica



DEITEL, M. D.; DEITEL, P. J.
Java, Como Programar
6a. Edição. Bookman., 2005.



ARNOLD, Ken; GOSLING, James; HOLMES, David
A linguagem de programação Java
4a. Edição. Bookman., 2007.



GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John
Padrões de projeto: soluções reutilizáveis de software orientado a objetos
Bookman., 2005.

Bibliografia complementar



GUEDES, Gilleanes T. A.

UML 2: uma abordagem prática

2a. Edição. Novatec., 2011.



SIERRA, K.

Use a Cabeça! Java

2a. Edição. O'Reilly., 2005.



DOWNEY, A. B.

Think Java: How to Think Like a Computer Scientist

5. ed. Green Tea Press, 2012.

Disponível em:

<http://greenteapress.com/thinkjava5/index.html>

Perguntas?

Alexandre de Andrade Barbosa

alexandre.barbosa@arapiraca.ufal.br